

A New Software Application for Footwear Industry

João L. Vilaça, Jaime Fonseca
University of Minho, Industrial Electronics Department
Guimarães, Portugal
Email: joaovilaça@dei.uminho.pt, jaime@dei.uminho.pt

Abstract – Today, the footwear industry is facing many challenges. First, consumers demand for new products with better comfort and design; second, competition is becoming stronger in current global market. Due to these factors, flexibility and rapidity in developing new products are key factors for the medium and long-term survival and success of the footwear industry. This paper proposes a new software application based in simple image processing techniques for optimization of two important steps of the processes involved in footwear manufacturing: the shoe sole halogenation and lead roughing process. The application presented in this paper has a friendly interface where the sole contour points for shoe sole halogenation and lead roughing are automatically determined. The operator can easily change and set new points to improve details within the interest region where tools will be applied, when the halogenation or the roughing process is executed. Another feature of this application is the automatic transformation of the 2D coordinates of the dominant points to 3D real world coordinates. This feature simplifies further ongoing work – automatic code generation for different industrial robots to execute the halogenation and roughing processes.

Keywords – Footwear industry, roughing process, sole halogenation.

I. INTRODUCTION

The global market is extremely demanding in terms of quality, flexibility and time response. The traditional footwear industries still get good level marks of performance; however, they need to increase its levels of automation to remain itself competitive. As effect, the companies of this sector need to change its organizational structure and at the same time to bring up to date the manufacturing technology, particularly in the connection between different productive chain stages, optimizing the flexibility and increasing the time response, as well as, in the quality control of the products and raw materials [1].

The work that will be presented here, it focuses in two different processes of the footwear industry: sole halogenation and lead roughing. These two processes are

always used in shoe manufacturing and their executions are all manual or a very simple semi-automatic system is used. The shoe sole halogenation is done before glue application. Its purpose is the preparation of the shoe sole to adequate adhesion of the glue. The chemical material used is toxic and harmful to operators. Roughing is the process that carries through the sandpaper of the shoe leather's zone (outside the sole), where later will be applied the glue and the shoe sole. [2].

Currently, the execution of these two processes is made in two ways:

- **Manual System** – Here, the two processes are completely manual. In this case the operator waits that a new sole or leather arrives to his place and executes the process manually with rudimentary tools (fig. 1).



Fig. 1 – manual system

- **Semi-automatic System** – Here, the operator is replaced by a robot. The execution mode depends on the program in the robot controller. This program reads a file containing a set of points; these points, represents the interest region where the robot executes the intended tasks (Fig. 2).

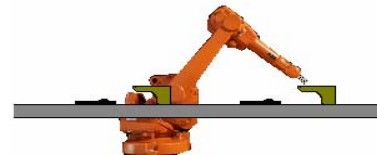


Fig. 2 – Semi-Automatic System

In this case, the points file is acquired by an off line software of teaching where a specialized operator select with a teaching tool a set of points. In this case, the points selected by the operator aren't, in the majority of the times, the correct ones. A good result depends a lot on the experience of the

operator. This needs to test the results a lot of times until the results are good enough, which represents a lot of time consuming. Relatively to the performance of the robot, here are some drawbacks:

1. The speed of execution depends of the number of selected points for the trajectory, and these are normally selected conservatively, so the running time is raised;
2. The quality of the roughing process depends on the position of each point, once the robot is programmed to execute linear movements between points. A bad choice in the point position can cause a cut in the leather and could damage the robot or the work space.
3. The robot doesn't differentiate the form of shoe present in the workspace. Always that a new shoe format is present, it's necessary to stop the robot and change the controller programming.

This paper presents a new software application (VisionFoot) based in image processing techniques to optimize these two important processes in the footwear industry. In the section two is present/discussed one proposal for a possible layout to be used with the VisionFoot in the footwear industry environment. A short description of the Chain code and Polygonal approximation methods used for the extraction of the dominant points of the shoe sole contour is described in section three. The algorithm implemented for extraction of dominants points in the shoe last, for both processes, is presented in section four. Practical results obtained from application on real cases of developed software are showed in the section five. Some conclusions about this entire new system concept are presented in section six. The future work is presented in section seven.

II. SYSTEM DESCRIPTION

In this section is proposed a possible layout to use in the footwear industry environment. This layout is divided in two levels: Office Level and Factory Level.

The Office Level is responsible for the digitalization and extraction of the dominant points that describe the interest region of the different shoe models and numbers of shoe forms manufactured in the industry. This level has an environment more clean and controlled than the Factory Level due the well known problems about the light interference in good image acquisition by the vision system. For image acquisition it is possible to use with the VisionFoot software a USB, FireWire or Composite Video Camera. The VisionFoot store all the information about the scanned shoes in a database. The information in the database is used at the Factory Level for code generation of the trajectories to the industrial robot controller. Next, the robot executes the halogenation or the roughing task.

The Factory Level is composed by an industrial robot, a rotating table with different shoe last and leads, an industrial

computer with TCP/IP communication for access to the database and an RFID equipment to identify the different shoes. Each different shoe form is identified by an identification code in the transponder. When the shoe form arrives at the robot work area, the RFID receiver detects the shoe form transponder and, reads it identification code. Then, the resident program that is running on the industrial computer analyzes the received code and sends the set of points stored in the database, associated with that shoe form, to the robot controller. When the robot receives the points, it starts to execute the two different processes, one at a time.

The layout approach of the system is show in the Fig. 3.

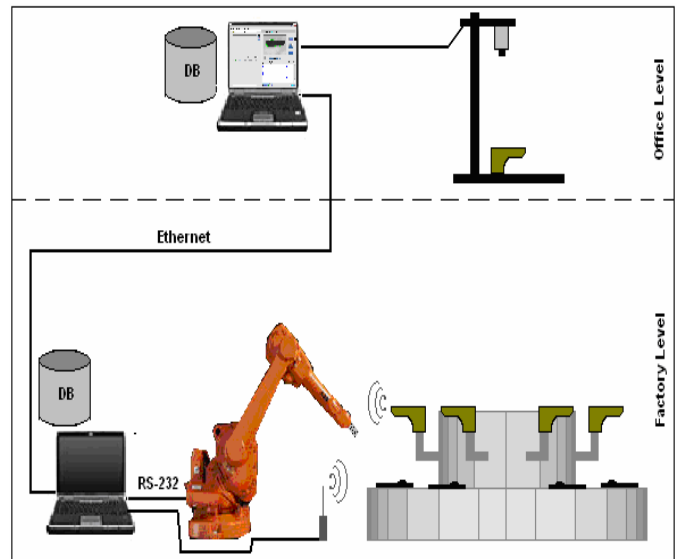


Fig. 3 – Automatic System, divided in office level and factory level.

III. CHAIN CODE AND POLYGONAL APPROXIMATION METHODS

In the VisionFoot are used three standard image processing techniques: Binarization, extraction of the contour – Freeman Chain Code [3] and the Polygonal Approximation [4] to select the dominant points. In this section, a summary on the Freeman Chain Code and the Polygonal Approximation is presented.

For contour-based shape representation and description it was select an 8-connectivity derivative Freeman chain code, which is based on the fact that an arbitrary curve is represented by a sequence of small unit length vectors and by a predefined set of possible directions. Through the encoding, successive contour points are connected to each other. The chain code uses a numbered sequence that represents relative directions of boundary points selected in a counter-clockwise 45° direction changes. The representation is based on 8-connectivity and the direction of each component is represented by the numbering scheme shown in Fig. 4.

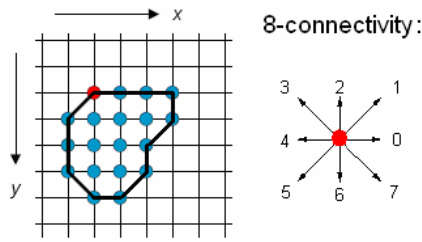


Fig. 4 – Chain Code method, with 8 – connectivity representation, chain-code started in the red point: 5 – 6 – 6 – 7 – 0 – 1 – 2 – 1 – 2 – 4 – 4 – 4.

The Freeman Chain Code result is a two dimensional space organized boundary of the segmented image. The boundary usually has thousands of points. Usually, having all the points are not useful, so, it was used a polygonal approximation method to reduce the number of points. The method is described in the present algorithm:

1. The user sets a tolerance distance (Td).
2. The algorithm traces a straight line between the initial (Pi) and the final (Pf) point of the contour (Fig. 5 a)).
3. The algorithm measures the distance between each curve point and the traced straight line, then it checks:
If the distance is greater than the tolerance (fig. 5 b)) – It creates two new straight lines, one between Pi and q (check point) and another between q and Pf.
If the distance is smaller than the tolerance – q is eliminated.
4. The algorithm moves to the next point Q1 (new check point), then it measures the distance between Q1 and the straight line q to Pf (fig. 5 c)), checking the condition like in point 3 of this algorithm.
5. The algorithm stops when q is equal to Pf.

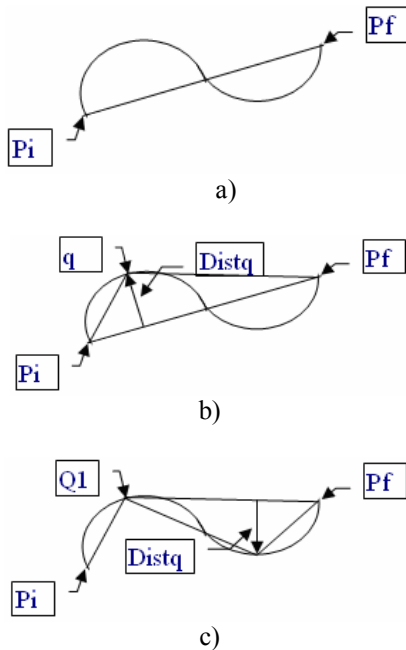


Fig. 5 – Polygonal Approximation method: a) initial step, b) recursive step, c) final step.

IV. THE VISIONFOOT APPLICATION

The VisionFoot was developed in Visual C++.NET [7], the image capture was made using DirectX SDK [5], this software allows us a capture from different kinds of devices. The human machine interface is showed in Fig. 7.

The main goal of this software is generate a set of real world points of the shoe form contour, and allow the use of this set of points directly in a robot program for the execution of the desired halogenation and roughing contour process.

As previously mentioned this software has two processes targets, one of the processes needs a contour inside the shoe form - halogenation, and the other, needs a contour outside of the shoe form - roughing. The Fig. 6 shows the software structure.

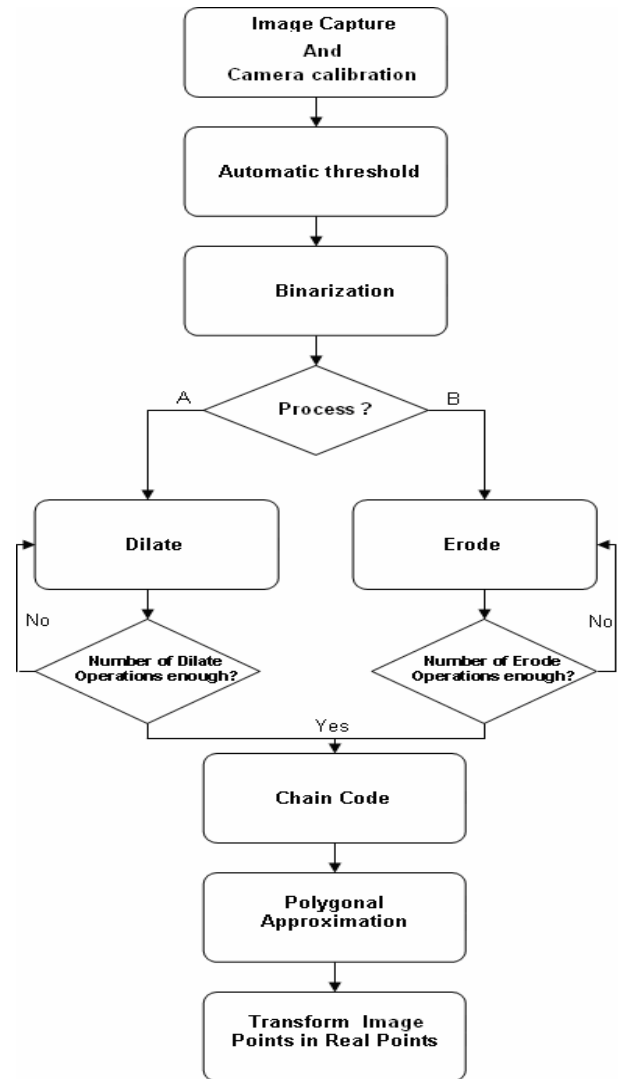


Fig. 6 – Software structure algorithm for the execution of the two processes.

The first step is the acquisition of an image from a camera, followed by a camera calibration. Camera calibration is understood as the process of determining the internal camera geometric and optical characteristics (intrinsic parameters), in

order to correct the image radial and tangential distortion [8] [9] [10].

The threshold level for the binarization process is automatically found based in the algorithm described by [6].

In the next step the user selects which process the robot is going to execute: halogenation or roughing process. In the first case, the main algorithm executes image erosion and in the second case executes an image dilatation. The user sets the number of times that each operation will be executed. The final image obtained is black and the remaining is white and is eroded or dilated relatively to the initial image.

Next, the main algorithm applies the chain code algorithm for extraction of the selected image boundary. This algorithm returns a set of 2D organized points of the shoe last contour. To minimize the number of contour points, without significant losses of information, the method of polynomial approach described previously was used. Before the application of this method the user needs to set the tolerance value of the contour. This tolerance is smaller in the roughing process.

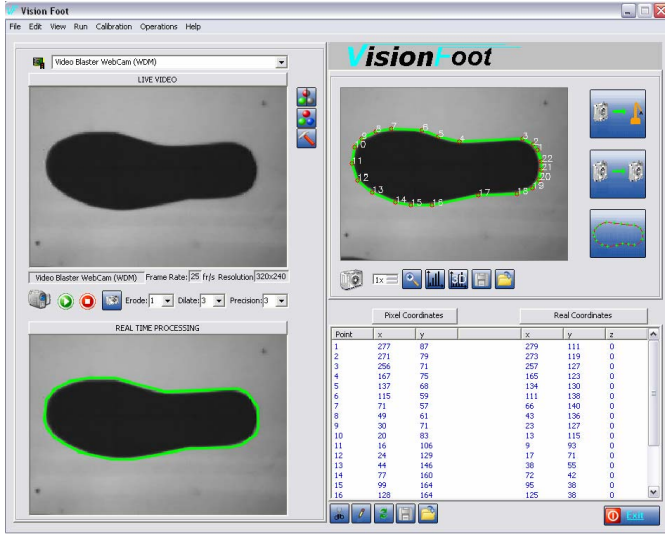


Fig. 7 – Software interface

Finally, the last step is to convert the contour set of dominant points in real world points. For that, the software has a wizard for image calibration. In this wizard the user selects four different points close to the image vertices.

For each point, the algorithm must match it to the real world coordinates. Then, for converting all points of the image contour to real world coordinates the program uses equation (1) for the vertical and equation (2) for the horizontal image contour points correspondence.

$$x_{Real}(x) = (x - P_{IC1} \cdot x) * \left(\frac{P_{IC2} \cdot x - P_{IC1} \cdot x}{P_{RC2} \cdot x - P_{RC1} \cdot x} \right) \quad (1)$$

With: $P_{IC1} \cdot x \leq x < P_{IC2} \cdot x$

$$x_{Real}(y) = (y - P_{IC1} \cdot y) * \left(\frac{P_{IC2} \cdot y - P_{IC1} \cdot y}{P_{RC2} \cdot y - P_{RC1} \cdot y} \right) \quad (2)$$

With: $P_{IC1} \cdot y \leq y < P_{IC2} \cdot y$

x – Image contour point vertical position.

y – Image contour point horizontal position.

PIC1 – Image pixel selected to be the initial point of the new coordinated system, for the vertical and horizontal component.

PIC2 – Image pixel selected to be the final point of the new coordinated system, for the vertical and horizontal component.

PRC1 – Real point value associated to PIC1, defined with zero millimeters in it vertical and horizontal component.

PRC2 – Real point value associated to PIC2, defined in millimeters in it vertical and horizontal component.

The VisionFoot application also allows a manual manipulation of the image contour points. The user can individually select each point of the image contour and move or erase it. The user also can create new points of the image contour.

The software also was 3D representation of the image plus the extracted contour points - 3D Histogram (Fig. 8). This represents an important user help in extracted contour visual verification. The 3D Histogram uses the grayscale image pixel position and intensity to build a 3D image. The 2D to 3D transformation is established by this relation: the x and a y pixel position represent the x and y 3D point positions and the z 3D point position value are represented by the pixel intensity. The 3D Histogram was developed using OpenGL Libraries.

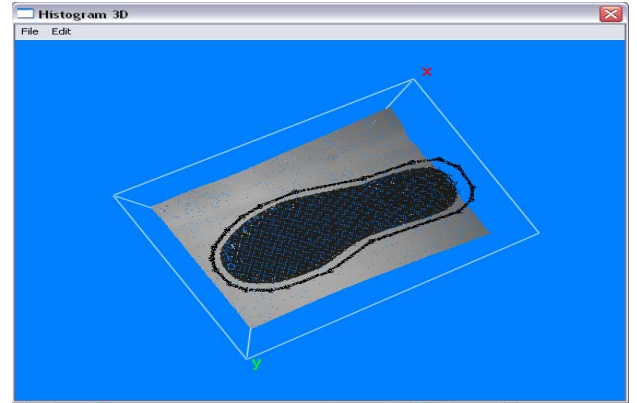


Fig. 8 – 3D Histogram based on grayscale information and pixels position, for 3D contour visualization.

V. RESULTS

This section presents the results obtained from the VisionFoot in the processing of a shoe last.

For software measurements validation, it has made a comparison between manual shoe last measures and software shoe last measures of the two processes extracted contour (Fig. 9).

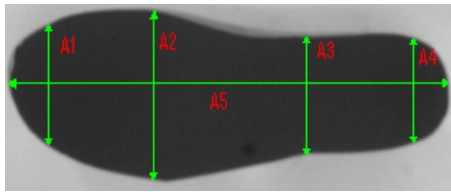
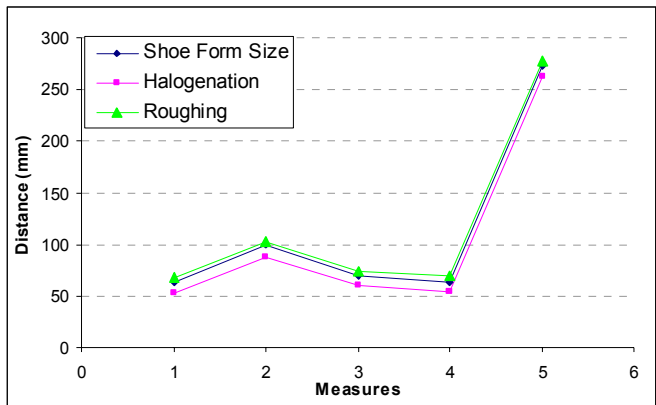


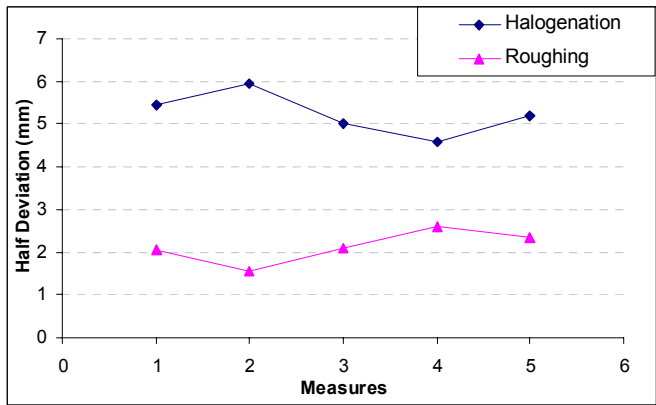
Fig. 9 – Measures executed on the shoe last.

TABLE I
MANUAL MEASUREMENTS VALUES VS SOFTWARE PROCESS MEASUREMENTS

	A1	A2	A3	A4	A5
Shoe last measurements (mm)	63.0	99.9	70.0	63.4	272.8
Halogenation Process measurements (mm)	52.1	88.0	60.0	54.2	262.4
Roughing Process measurements (mm)	67.1	103.0	74.2	68.6	277.5



a)



b)

Fig. 10 – Measurements made on five different spots: a) measures values in the shoe last, software Halogenation contour and software roughing contour, b) half deviation value of the software Halogenation and roughing contours with reference to the shoe last measures.

From the results it is possible to verify that the roughing measures values are greater than the measures obtained manually over the shoe last and that the halogenation measures values are smaller than the measures obtained manually over the shoe last.

These differences allow the system to drive the process in the desired way.

The differences between measurements are dependent on the number of the dilatations and erosions operations. The number of dilations or erosions depends on three factors: the distance between the camera and the shoe sole, the camera lens and the camera resolution. In these tests it was used a camera with a resolution of 320x240, a lens of 8 mm and the distance between the camera and the shoe sole it was 700 mm.

With this image acquisition conditions the number of erosions (Halogenation process) and dilatation (roughing process) are respectively five and two.

The processing time in this test was 100 milliseconds. With this processing time it is possible to assume the use of this software in online identification of shoe last.

VI. CONCLUSIONS

The VisionFoot concept presented here it is not finished but presents now a set of characteristics that distinguishes it from current solutions:

- Independence between system operational modules;
- Friendly man-machine interface;
- Flexibility;
- Increased productivity;
- Reduced dead times;
- Improved quality processes (Halogenation and Roughing);

In spite the fact that the system has a functional independence between the two modules, this system works as an integrated system. Therefore, it is the integration of factory and office modules that distinguishes this solution and promotes the productivity increase and dead times reduction.

It presents a simple and intuitive interface, becoming unnecessary the presence of specialized operator. The Vision Foot is very flexible; it is quite easy to insert a new shoe sole or a new shoe last with different sizes, as well as, editing existing models in the shoe database.

Based in the analysis of the results it is possible to conclude that, VisionFoot application significantly improves the two processes quality, namely in the cleanness shoe sole with the toxic product and the leather roughing.

VII. FUTURE WORK

In the future work, the integration between the two layout levels, it will be developed. Developing a new concept, where the recognition, extraction of dominant points and the

execution of the process will be carried out at the same level – factory level, without human intervention.

Another, functionality in development is the automatic code generation for different industrial robots based in the points extracted by the VisionFoot software application.

REFERENCES

- [1] Manuel J. Ferreira, Aparício Fernandes, J.A. Campos Neves, João L. Monteiro, *"Automatic Visual Inspection on Shoe and Textile Industries"*, IASTED CGIM'98, Halifax - Canada, 1998.
- [2] Sepúlveda João, Couto Carlos, Monteiro João L, *"Developing an automated system for shoe sole halogenations"*, ISIE'97. IEEE INTERNATIONAL SYMPOSIUM ON INDUSTRIAL ELECTRONICS, Guimarães - Portugal 1997.
- [3] Herbert Freeman, *Computer processing of line-drawing images*. Computing Surveys, 6(1):57-97, March 1974.
- [4] M. DeHaemer, Jr. and M. J. Zyda. *"Simplification of objects rendered by polygonal approximations"*. Computers and Graphics, 15(2):175 – 184, 1991.
- [5] Mark D. Pesce, *Programming Microsoft Directshow for Digital Video and Television*, Microsoft Press, 2003.
- [6] John C. Russ, *The Image Processing Handbook Third Edition*, CRC Press, 2000.
- [7] John E. Swanke, *VC++ MFC Extensions by Example*, CMP Books, 1999.
- [8] R.Y. Tsai, *"A Versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses"*, IBM Res. Rep. RC 51342, May 8, 1985.
- [9] Heikkilä, J., Silven, O., *"A four-step camera calibration procedure with implicit image correction"*, Proc. CVPR '97, IEEE, 1997, pp. 1106-1112.
- [10] Vilaca, Joao L.; Fonseca, Jaime; Pinho, A. C., *"Stereo Vision Calibration Procedure for 3D Surface measurements"*, IEEE Industrial Electronics, IECON 2006 - 32nd Annual Conference on, Vol., Iss., Nov. 2006 Pages:3508-3513.